REMARKS/ARGUMENTS

Favorable consideration of this application is respectfully requested.

Claims 1-20 are pending in this application. Claim 20 has been amended as suggested at the middle of page 3 of the outstanding Action. Accordingly, as the suggestion in the outstanding Action has been adopted, it is believed to be clear that no new matter has been added.

In addition, as the present Amendment simply adopts the suggestion in the outstanding Action to reduce issues on appeal, it is believed that the present amendment that requires no new search and that creates no other examination burdens should be entered under 37 CFR § 1.116.

The outstanding Official Action presented a rejection of Claims 20 under 35 U.S.C. §101 again asserting that this claim is directed to non-statutory subject matter and a rejection of Claims 1-20 as being anticipated by Iline et al. (U.S. Published Patent Application No. 2004/0068491, Iline).

The rejection of Claims 20 under 35 U.S.C. §101 is believed to be overcome by the present amendment that adopts the suggestion at the middle of page 3 of the outstanding Action in terms of adding the language " loaded in a processor" after each recital of a "computer program code." Accordingly, withdrawal of this rejection of Claim 20 under 35 U.S.C. §101 is respectfully requested.

Before considering the outstanding prior art rejection based upon anticipation of Claims 1-20 by Iline, it is again believed that a brief review of the present invention would be helpful. In this regard, it is noted that independent Claims 1, 19, and 20 clearly require producing a copy of the hierarchal data at a time of starting an access to the hierarchical data as to each transaction. Writing access as to each transaction is then made as to its respective copy, while avoiding a collision with accesses as to other transactions. Thus, the present

invention includes three primary technical elements in terms of requiring i) hierarchal data; ii) functions or commands to access this hierarchal data; and iii) a procedure for processing a plurality of the transactions that includes the improvement that avoids collisions.

While Iline also teaches, i) hierarchal data and ii) functions or commands to access this hierarchal data, the thrust of Iline is to use a restricted acess model data structure (e.g., like that in FIG.4) to implement the functions or commands to access this hierarchal data. There are no teachings or suggestions of the third element (i.e., a procedure for processing a plurality of the transactions that includes the improvement that avoids collision).

By virtue of the suggested use of object-oriented programing, the Iline restricted access model data structure facilitates protection of data, while allowing access to that data when needed. Thus, malicious client code will be restricted with respect to its ability to corrupt stored data, see paragraph [0027], for example.

On the other hand, the present invention is concerned with providing a procedure that permits procesing a plurality of transactions in parallel, not with the Iline restricted access model data structure that facilitates protection of data, while allowing access to that data when needed. These two approaches are different and they are directed to different levels of database technology.

Page 12 (at lines 14-20) of the outstanding Action suggests that Iline teaches "instantiating a data structure object through one or more constrictors" as to paragraph [0042] and that this corresponds to the Claim 1 step (and similar limitations in Claims 19 and 20) requiring "producing a copy of the hierarchical data at a time of starting an access to the hierarchical data by each transaction." In this regard, the actual teaching of paragraph [0042] is that "the restricted access model data structure is obtained (Step 160)." It is with regard to this obtaining of the "the restricted access model data structure," not any "starting of an access to the hierarchal data," that this paragraph teaches "the restricted access model data

structure may be obtained by instantiating a data structure object through one or more

constructors." Thus, as one or more of these "constructors," paragraph [0042] then

references "[a]n initial constructor, in accordance with one or more embodiments of the

invention, may receive one or more parameters, e.g., a multi-part data structure, such as a

series of linked vectors or arrays, where a first element of the data structure (e.g., a first

vector) is linked to a second element of the data structure (e.g., a second vector)."

It is clear that paragraph [0042] is merely indicating that when instantiating a data

structure object, hierarchal data is copied from a source multi-part data structure. However,

this copying is clearly being done to set up ("obtain") the "the restricted access model data

structure," not "at a time of starting an access to the hierarchical data by each transaction."

In accordance with Iline, an instance object of TestData is created with a new

DataReader of "Root" associated with DataWriter of "Root." Then, a pair of instances of

DataReader and DataWriter can be repeatedly created by invoking DataWriter add in order to

establish the hierarchal data structure. The set of instance objects of DataReader and

DataWriter need not be written back unless the user wants to review the data.

If each transaction is serially performed, i.e., only after the preceding transaction has

been completed, the hierarchal data structure is accessed without problem. Conversely, if

transactions are performed in parallel, an appropriate scheduling or schema necessary for

allowing parallel transactions must be introduced which is not disclosed by Iline. In this

regard, the present invention is different from Iline.

The top of page 13 of the outstanding Action states that:

> **Writing access as to each transaction is then made as to in respective copy, while avoiding a collision with accesses as to other transactions** is equivalent with *TestResultWriter, TestResultReader (See Code Listing A in page 4, 0044).*

However, DataReader implements TestResultReader without a reference to

DataWriter implementing TestResultWriter, in order that one node is prevented from writing

data to another node. This is the gist of Iline, that is, protection of data against inter-node

writing. On the other hand, the present invention is related to protection of data associated

with one transaction against collision from parallel transactions. With regard to the present

invention, it depends on the actual case whether of not inter-node writing is undesirable. In

some cases, inter-node writing might be allowed. However, protection of data against

collision of parallel transactions is always implemented to prevent collisions when a plurality

of transactions are processed in parallel. The present invention is completely different from

Iline in this regard.

The middle of page 13 of the outstanding Action notes that:

B. The design of Iline insures the no collision between
accesses made by different reader-writer pairs could possible occur.

Iline teaches this limitation as follows:

**collisions between accesses made by different reader-writer
pairs could possible occur** corresponds to *The writer interface does not
have a method available to alter a data already stored in the restricted
access model data structure;* **corrupting the data** *([0041]).*

**collision between accesses made by different reader-writer
pairs could possible occur** is equivalent with *TestResultWriter;
TestResultReader (See Code Listing A in page 4, 0044).*

As has been discussed above, there is no description of collisions in Iline. Collisions

occur at the same location rather than between different nodes. Iline seeks to prevent one

node from writing another node even at separate timings, i.e., even when transactions are not

concurrently processed but only serially processed, i.e., one after another. This is prevention

of corruption, not prevention of collision. In this regard, it is only if there are plural

transactions at the same node and at the same time that collision occurs. This is true even in

the restricted access model data structure taught by Iline. There is no countermeasure to

prevent such a collision taught by Iline.

The bottom of page 13 through page 14 notes that:

       C. Iline does not teach or suggest judging whether any collision occurs and then carrying out a processing for avoiding the collision.

On the contrary, Iline reads on the claimed limitations as follows:

       Judging whether any collision occurs corresponds to *relationships between data may be determined by the formatter (32), the formatter (32) may then format the data using the relationships ([0041]),*

       then carrying out a processing for avoiding the collision corresponds to *then format the data using the relationships ([0041]).*

       However, paragraph [0041] simply states that "[b]ecause relationships between data may be determined by the formatter (32), the formatter (32) may then format the data using the relationship." This statement simply means that the formatter formats instance object data according to XML, and the like, for display. The relationships to be determined for formatting are relationships among the members of TestResultWriter and TestResultReader in the hierarchical model.

Page 14 of the outstanding Action finishes by noting that:

       Any collision occurs corresponds *to the writer interface **does not have a method available** to alter a data already stored in the restricted access model data structure; **corrupting the data** ([0041]).*

Considering the technical concept of <u>Iline</u>, this sentence apparently suggests that the data associated with one location (or node) shall not be altered by another location (or node) to avoid corrupting the data. But data corruption prevention is not collision prevention as fully explained above.

Consequently, <u>Iline</u> actually fails to disclose all the features recited in independent Claims 1, 19 and 20 and therefore, it is logically impossible to suggest that <u>Iline</u> anticipates Claims 1, 19, and 20. Accordingly, withdrawal of this rejection based on anticipation of Claims 1, 19, and 20 is respectfully requested.

Further, as Claims 2-18 depend from Claim 1 and include all the subject matter thereof, these claims clearly patentably define over <u>Iline</u> for at least the reasons set forth above as to Claim 1. In addition, each of these dependent claims adds features that are

further not taught or suggested by <u>Iline</u> and patentably define thereover for this reason as well. Therefore, withdrawal of the rejection of Claims 2-18 is also respectfully requested.

Accordingly, it is respectfully submitted that no further issues remain outstanding in the present application, and that this application is clearly in condition for formal allowance and an early and favorable action to that effect is, therefore, respectfully requested.

Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Eckhard H. Kuesters
Attorney of Record
Registration No. 28,870

Raymond F. Cardillo, Jr.
Registration No. 40,440

Customer Number

22850

Tel: (703) 413-3000
Fax: (703) 413 -2220
(OSMMN 06/04)

I:\ATTY\RFC\24\248156-AM.DOC

15